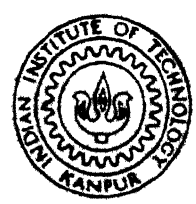
ANGLABHARTI :
A MACHINE AIDED TRANSLATION SYSTEM FROM ENGLISH TO INDIAN LANGUAGES- ENGLISH TO TAMIL VERSION

By

SIVARAMAN K.



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

APRIL, 1993

ANGLABHARTI:
A MACHINE AIDED TRANSLATION SYSTEM FROM ENGLISH TO INDIAN LANGUAGES - ENGLISH TO TAMIL VERSION

A thesis submitted
in partial fulfilment of the requirements
for the degree of
Master of Technology

by
SIVARAMAN K.

to
THE DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
KANPUR - 208 016

APRIL 1993

Certificate

Certified that the work contained in this thesis titled **"ANGLABHARTI: A Machine Aided Translation System from English to Indian Languages - English to Tamil Version"** has been done by **Sivaraman K.** (Roll No. 9111129) under my supervision and it has not been submitted elsewhere for a degree.

Dr. R.M.K. Sinha
Professor and Head
Dept. of Computer Sc. & Engg.
I. I. T. Kanpur

21 April, 1993

CSE-1993-M-SIV-MAC

ACKNOWLEDGEMENT

My thesis will not be complete without my deep sense of gratitude towards my thesis supervisor Dr. R.M.K. Sinha, who, in spite of being one the busiest persons I have ever met, devoted considerable portion of his time towards making my thesis a presentable one. I thankfully acknowledge his valuable guidance, help and crucial insights during the course of this thesis.

Full credit to the MAT - ANGLABHARTI team under Dr. R.M.K. Sinha, for their valuable assistance from time to time. Special thanks to Miss Aditi Agarwal, whose valiant efforts had saved my implementation repeatedly.

I thank my Tamil friends, classmates, critics of this work and the gregarious comrades of B-Top/5 whose company I cherish.

Sivaraman Krishnamurthy.

Abstract

ANGLABHARTI is a Machine Aided Translation System from English to Indian Languages. This work deals with system design aspect of ANGLABHARTI, with specific reference to English to Tamil translation.

ANGLABHARTI exploits the commonality among Indian Languages to obtain a pseudo-intermediate representation from English. The computational effort in arriving at this pseudo-intermediate representation is kept minimal. This makes the system more practical. The system envisages the use of human assistance to improve the quality of translation.

The structural transformation is taken care of by storing prominent patterns in English and their associated transformations in the target languages, using a grammar.

The semantic disambiguation is made variously using patterns, syntactic attributes and selectional restrictions, enforced by semantic tags.

All the text generators for various target languages use the same intermediate output generated by the preceding stages of ANGLABHARTI. This way a generic translation system for Indian Languages is obtained.

Contents

Chapter 1

Overview of ANGLABHARTI

1.1 On Machine Translation

The task of MT can be defined very simply: the computer must be able to obtain as input a text in one language (SL, for source language) and produce as output a text in another language (TL, for target language), so that the meaning of the TL text is the same as that of the SL text [31]. This leads to a number of questions:

1. What is the meaning of the text?
2. Does it have any component structure?
3. How does one represent the meaning of a text?
4. How does one set out to extract the meaning of a text?
5. Is it absolutely necessary to extract meaning in order to translate?

All the above problems are difficult. Multiple senses lead to the problem of **disambiguation.** A significant amount of semantic and pragmatic analysis of natural language is required before the disambiguation can be achieved.

There are two major avenues of circumventing the problem of completely automatic disambiguation. First, one can restrict the grammar and the vocabulary of the input text in such a way that most of the ambiguity is eliminated. This is the *sublanguage*, or *subworld*, approach to MT. Second, one can drop the requirement of complete automation and allow humans to get involved in the translation process. This is *Machine-Aided Translation* approach. The difference between these approaches is not only in the tactics of interspersing automated and

manual steps in the process of translation, but also in the nature of the subtasks for which humans are responsible.

ANGLABHARTI falls under the second category.

1.2 On Human Interaction in Machine-Aided Translation

With respect to the strategy of human involvement in MAT, there are three possibilities: *pre-editing*, *post-editing* and *interactive editing*. A human pre-editor reads the input text and modifies it in such a way that the MT system is able to process it automatically. Difficult and overly ambiguous words and phrases are replaced with those that the editor knows the program will handle. A human post-editor, conversely, obtains the output from an MT system and eliminates all inaccuracies and errors in it. An interactive editor engages in a dialog with the MT system, in which the human resolves ambiguities that the machine is not capable of resolving itself. It is, of course, necessary to build a special interface to maintain this kind of dialog.

ANGLABHARTI employs post-editing predominantly, and to a lesser extent, interactive-editing.

Further information on human-machine interaction in translation may be obtained from [29].

1.3 Strategies in Machine (Aided) Translation

Three major strategies have governed the design on MT systems over the last two decades [40], viz. *Direct Translation* strategy, *Transfer* strategy and *interlingua* strategy.

The direct translation system is designed, from its outset, for a specific source and target language pair. No general linguistic theory or parsing principles are necessarily present for direct translation to work; these systems depend instead on well-developed dictionaries, morphological analysis, and text processing software to gain credible translations of the source text into a series of reasonably equivalent words and phrases in the target language. SYSTRAN system [38] is an example.

In the transfer strategy, a source language (SL) sentence is first parsed into an abstract internal (usually, some sort of annotated structure) representation. Thereafter, a 'transfer' is made at both the lexical and structural levels into corresponding structures in the target

language (TL). In the third stage, the translation is generated. Three dictionaries are needed for transfer: an SL dictionary, a bilingual transfer dictionary, and a TL dictionary. The level of transfer differs from system to system - the representation varies from purely syntactic deep structure markers to syntactico-semantic (compositional semantics, case frame information, and so forth) annotated trees. Note that the transfer stage involves a bilingual component, i.e. one tailored for a specific SL-TL pair. This strategy was popularised by system like SUSY [26].

An alternative approach is to develop a universal, language-independent representation for text, known as *interlingua*. Here the MT model has two phases: analysis and generation. In principle, we can dispense with bilinguality. For a multilingual system with n SLs and m TLs the transfer approach will require mn transfer blocks (if the sets of SLs and TLs are disjoint), in addition to n analyzers and m generators. In the interlingua approach, only n parsers and m generators will be needed. This uses the AI approach.

The strategy in ANGLABHARTI is better than the transfer approach, as the translation is valid for a host of TLs, but falls short of a genuine interlingua, in that it ignores the meaning of the text to be translated.

1.4 Design Objectives

1. The primary motive of this system is to exploit the closeness exhibited by the family of Indian languages. A simple paradigm is devised which is good enough to translate from English to an intermediate form, using which different target text generators construct the translated output in Indian languages.

2. The unit of translation is a sentence. No attempt is made to incorporate intersentential context.

3. The system is not guaranteed to produce 100% perfect translation all the time. A human postediting after the machine translation is not ruled out. The approach is based on heuristics, using A.I. techniques.

4. Translation with minimal understanding of the text. Many of the MT systems have proved to be not viable for general purpose translation, simply because they require a deeper analysis of the sentence. The present approach requires simpler processing of the given sentence in English. In fact, the paradigm involves transferring the surface

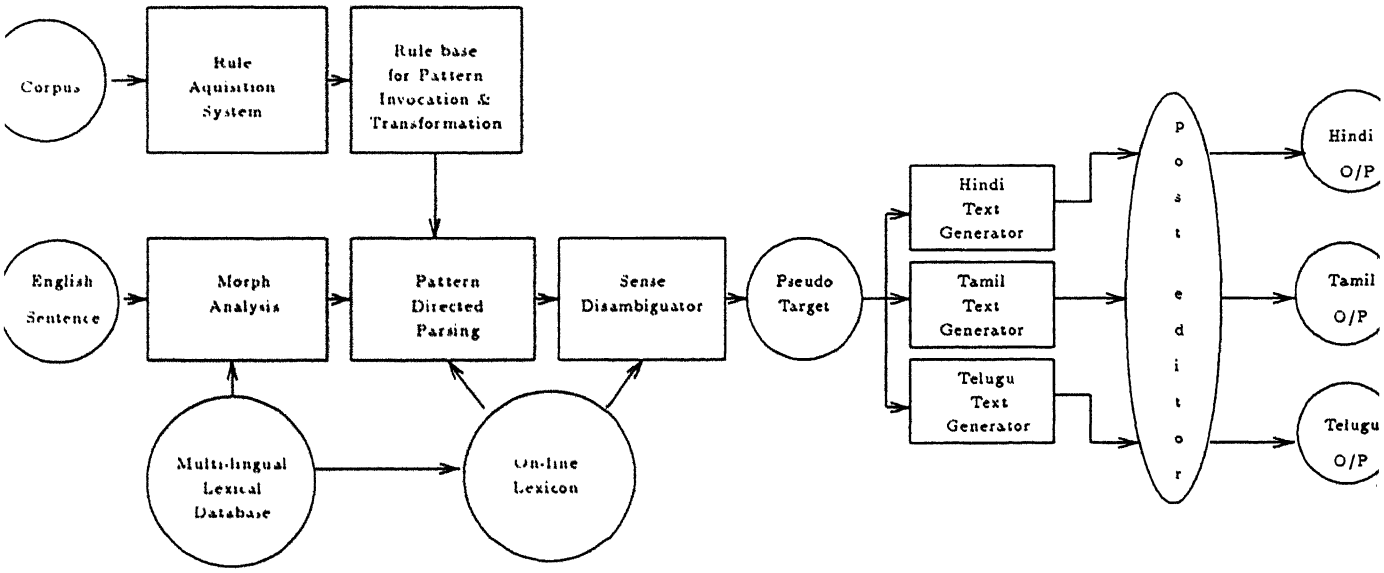


Figure 1.1: Block Schemata of ANGLABHARTI

structure from English to that in Indian languages, without penetrating into the *deep structure* [11] of the sentence.

1.5 System Components

The following are the major components of ANGLABHARTI:-

1. *Rule base.* This contains rules for mapping structures of sentence from English to Indian languages. This database of pattern-transformations from English to Indian languages is entrusted the job of making a surface-tree to surface-tree translation, bypassing the task of getting a deep tree of the sentence to be translated.

2. *Sense disambiguator.* This module is responsible for the picking of the correct sense of each word in the source language. It should be of interest to note that sense disambiguation is done for only the source text.

   The approach used in ANGLABHARTI may be termed *rule-by-rule semantic interpretation* [2, 3]. Here the semantic interpreter is called each time a syntactic rule is applied.

3. *Target text generators.* These form the tail ends of the system. Their function is to generate the translated output for the corresponding target languages. They take as input the intermediate form generated by the previous stages of ANGLABHARTI. Note that their task is quite different from what is called *Natural Language Generation* [13], in that the latter has also to decide 'what to say' (the strategic level) in addition to 'how to say it' (the tactical level).

   Note that by having different text generators using the same rule base and sense disambiguator, a generic MT system is obtained for a host of target languages.

4. *Multi-lingual dictionary.* This contains various details for each word in English, like their syntactic categories, possible senses, keys to disambiguate their senses, corresponding words in target languages.

5. *Rule base Acquirer.* This prepares the rule base for the MT system. This module involves a suitable Machine Learning paradigm.

Figure 1.1 shows the interaction among the components mentioned.

A similar Machine Translation System using structural transformation, followed by sense disambiguation between English and Japanese is *the Mu-Project* [39].

1.6 Rule Base

English is a verb-central language, whereas the Indian languages may be treated as verb-final ones. Some of the characteristics of verb-final languages include: [27]

1. Verb is typically the last word.
   eg. avan kadaikku chenrAn
   he shop -to go
   he went to the shop
2. Postpositions after nouns. eg. magan -ikkAga (son -for)
3. Modifiers precede the head noun.
4. Auxillaries follow main verb.

A set of pattern directed rules is constructed, which transforms the surface structure of the sentence in English to an intermediate form.

A typical movement rule is:

noun-phrase verb-phrase prep-phrase → noun-phrase′ prep-phrase′ verb-phrase′

eg. I am going to the market → intermediate_form

intermediate_form = [noun,dont_care,singular,first,(HUMAN,ANIMATE),I,nAn,main],
[noun,neuter,singular,third,(PLACE),market,kadai,bazAr],
(prep,to),[am_verb_5,go,(pO,pO),(jA,jA)]

This is used by the Text generators to output:

(Tamil) nAn mArkettukku pOgirEn
(Hindi) main bAzAr jA_rahA_hUng

Such movement rules are identified to construct rule base. The construction of rule base in ANGLABHARTI was done after analysing the possible patterns in English of oft occurrence **[1, 18, 33, 34 , 44]**.

The idea of using structural transformations in MT is quite common. Makoto's system[**30**] uses similar ideas to translate between English and Japanese.

1.7 Sense Disambiguation

The ambiguity resolution is predominantly carried out using semantic tags. Detailed discussion on the mechanism used can be found in subsequent chapters.

1.8 Dictionary Organisation

A two-tier organisation of lexicon is suggested. The external lexicon has the standard information about the words. The "*on-line lexicon*" has information about only the words currently encountered in the source sentence. This scheme helps in simplifying details of the parser used for movement rules. Typically root words are stored in the external lexicon, whereas the full form of the word is put in the on-line lexicon.

Elaborate details about the storage of idioms and phrases are also considered to provide a general purpose translation.

1.9 Text generation

The different text generators for Hindi, Tamil, and Telugu use information like the morphology of root words, special properties of categories and other related details necessary for the individual target language. It is of interest to note that even in this last stage of ANGLABHARTI considerable similarity is exhibited in the working of different text generators.

Elaborate discussion of Tamil Text Generator can be found subsequently.

Chapter 2

Movement Rules

2.1 Movement Rules

The idea of using surface patterns to capture the meaning of sentences is quite old in linguistics [17]. Although simple, this scheme is effective in capturing the idiosyncrasies of the surface patterns of a language.

The phrase structure grammar of Chomsky [8] and the c-structure of lexical-functional grammar [20] has a lot in common with the approach used in ANGLABHARTI.

The data base of structural transformation rules from English to Indian languages, hereafter referred to simply as rule base, forms the heart of ANGLABHARTI system. This takes care of the crucial changes in the syntax while translating from English. As mentioned earlier, by making a generic rule base for Indian languages, ANGLABHARTI exhibits a potential benefit while translating from English.

The subsequent sections discuss the various concepts regarding the rule base.

2.2 Components of rule base

The following are the major components of rule base:

- *Phrases.* Typical word units like noun-phrase, verb-phrase and prep-phrase.
- *Case markers.* These are the units that express an implicit semantic relationship in English pattern, which however has to be explicitly denoted in Tamil.

eg. noun-phrase-1 verb-phrase noun-phrase-2 → noun-phrase-1′ noun-phrase-2′ k1 verb-phrase′

I called her → nAn aval -ai kUppittEn

Here k1 is mapped to the Tamil suffix '-ai', denoting that noun-phrase-2 serves as an object to verb-phrase.

- *Literals.* There may be literals in the movement rules of rule base. They are to be interpreted by the target text generators accordingly.

  eg. sentence-1 and sentence-2 → sentence-1′ l1 sentence-2′

  Here l1 is a literal interpreted as 'matrum' by Tamil text generator and 'Owr' by Hindi text generator. This helps in constructing a generic rule base.

- *Parameter mechanism.* When the word units are moved, the other units with which they are associated has to be specified. For instance, when verb-phrase is moved, its parameters like tense, modality and the noun-phrase to be used for gender-number-person (gnp) agreement are specified as parameters to the verb-phrase mapper.

- *Macros.* In order to enable the use of the same rule base for translation into a variety of Indian languages, the idea of macros is introduced.

  eg. noun-phrase whose sent-1 rest-of-sent → whose(noun-phrase,sent-1) rest-of-sent′

  Here whose() is a macro that specifies how its parameters are to be modified according to the specific target language. Such macros are embedded in target text generators.

  Consider the noun phrase: the lady whose bag was stolen, which is parsed as whose(noun-phrase(the lady), sent-1(bag was stolen)). Now the Tamil text generator expands this as:

  enda noun-phrase′(the lady) -udaiya sent-1′(bag was stolen) -O anda noun-phrase′(the lady)

  whereas the Hindi text generator expands this as:

  jis noun-phrase′(the lady) -kA_form sent-1′(bag was stolen) vus noun-phrase′(the lady)

2.3 Organisation of rule base

Due to the large rule base necessary for any decent translation, the efficiency of the parser suffers. Hence the rule base must be organised suitably.

The following gives a guideline to how the rule base is organised:

- Each rule has a pattern occurring in English and a corresponding pattern in Indian Languages, expressed in terms of the components in English pattern.

- The rule base itself is hierarchically structured. There are separate rule bases for noun-phrases, verb-phrases, prep-phrases etc. In addition, there is a rule base for sentences that are expressed in terms of the other lower level rule bases. This is because the movements within the individual phrases during transformation is largely independent of the surrounding word-units.

  Suppose there is a pattern 'det-star adj-star noun' for noun-phrases. This pattern is translated in Tamil as 'det-star adj-star noun' independent of the other components of the sentence with this noun-phrase.

  Thus by using this hierarchy of rule base, the number of sentence-patterns is effectively reduced. Where such independent transformations within a hierarchy is not possible (for instance, when a noun-phrase cannot be transformed independent of its relative clause, and the target language for translation), one can use the facility of literals and macros within the rule base, as explained already.

- Only the English patterns for simple sentences are captured in rule base. All other variations are expressed in terms of the existing rule base for simple sentences and individual phrases. This further reduces the size of rule base. This is in tune with Chomsky's view [8] "the grammar of English is materially simplified if phrase structure description is limited to a kernel of simple sentences from which all other sentences are constructed by repeated transformation".

- Where it is convenient to use a special form of an existing pattern, it is entered in rule base separately.

  For instance, it is convenient to treat 'noun-phrase-1 is noun-phrase-2' pattern as distinct from 'noun-phrase-1 verb-phrase noun-phrase-2' pattern, as the latter typically requires a case marker in the corresponding target pattern.

- Slight deviation of an English pattern is taken care by making use of the existing rule base differently, according the need.

  Consider the noun-phrase pattern, 'noun-phrase who slot'. Here 'slot' stands for a sentence pattern, excepting that the subject will be missing, as in 'the girl who became angry'. It is prudent to use the existing rule base for such variations. ANGLABHARTI permits the same.

2.4 Principle of operation

The following are some of the features that enable the scheme mentioned to be effective.

- There is a strong correlation between syntax and semantics in English. This enables one to ascertain the semantic role of a word unit from the syntax of the sentence. For example, the surface pattern 'noun-phrase-1 verb-phrase noun-phrase-2' can be interpreted to mean 'subject action object'. This means the case relationship can be identified from the pattern itself.

  Note that this view is quite familiar in linguistics. Chomsky has repeatedly claimed **[9, 10]** that grammatical relations such as subject-of and object-of can be equated with configurational relations in the deep structure.

- Among the Indian languages, similar grammatical properties can be identified. In particular, while translating from English, the movement rules do not differ widely. This enables building a generic rule base, as mentioned earlier.

- Due to the human engineering involved in constructing the rule base, the translation can be easily tailored to fit the styles of target languages.

- In any MT system, two major components are involved: the movement of word units and the sense disambiguation of word units. Because a viable scheme for sense disambiguation with minimal understanding is identified (to be elaborated later), the above rule base technique can be adopted.

- ANGLABHARTI ignores intersentential context. Hence the structural transformation rule for an isolated sentence can be used.

Now some specific trouble-shooters in this technique are identified, together with suggestions about overcoming them.

2.5 Rule base acquisition

A major trouble with the above approach is in acquiring the rule base. English is a rich language with an enormous variety of patterns. Clearly, a universal set of patterns is yet to be identified.

Hence the rule base cannot be static. It should be augmented by a module that acquires new rules, using a suitable machine learning paradigm. Different paradigms for acquiring such structural transformation rules can be obtained from [6].

It is of interest to note that acquisition of rule base from examples strongly resemble the *projection problem* [21]. A speaker's knowledge of his language takes the form of rules which project the finite set of sentence he has fortuitiously encountered to the infinite set of sentences of the language. A description of the language which adequately represent the speaker's linguistic knowledge must, accordingly, state these rules. The problem of formulating these rules is referred to as the projection problem.

Alternatively, the rule base may be manually updated periodically depending on its performance, an approach currently followed in ANGLABHARTI. In the developmental stages, this is one of the best approaches possible.

Clearly the perfect translation of any text is not feasible without involving a deep understanding. The technique presented here is meant only to tackle the commonly used text formats. It is believed that the cost involved in any further refinement increases rapidly, and hence may not be worthwhile if a human engineering is envisaged for the post-editing of the translated output.

2.6 Ambiguity resolution

The use of pattern-matching technique leads to the problem of resolution in case of contentions. It is very likely that two or more patterns fit the given sentence. In tune with the spirit of minimal understanding, the resolution is carried out by studying the various conflicts and encoding the resolution in the rule base for each possible contention.

For instance, consider the word 'her'. ANGLABHARTI treats this both as a possessive case and as a noun. The ambiguity is resolved by taking it to be the former whenever it is followed by a noun, and as the latter, otherwise.

As an another example, consider the sentence "flying planes is dangerous". Here the noun-phrase 'flying planes' fit the pattern 'adjective noun' as well as 'gerund noun'. However, the rule base rejects the first pattern as it mismatches with the expectation of the verb 'is' in the sentence.

2.7 Identification of case markers

As explained already, some of the target patterns involve the identification of suitable case markers. For instance,

noun-phrase-1 verb-phrase noun-phrase-2 → noun-phrase-1′ noun-phrase-2′ k1 verb-phrase′

Here k1 is a case marker which is to be identified for each target language by the corresponding text generator. However, the case marker k1 for a given language cannot be fixed from the surface pattern itself.

Illustrating with examples for the target language Tamil,

Rama called Sita → Rama Sita -ai kUpittAn (k1: -ai)

Rama went home → Rama vIdu -ikku pOnAn (k1: -ikku)

Currently work is underway in resolving the ambiguities in case marker mappings in target languages.

2.8 Merits of the scheme

In spite of being a shallow approach, the scheme merits some consideration.

1. Resolution of parts of speech is avoided. In any MT system, it is imperative that the parts of speech of each word in the sentence be identified. Several words can function in multiple parts of speech, for instance 'bark' can be 'noun' or 'verb', the resolution of which requires inquiring into the role played by them. However by virtue of using surface templates, the parts of speech is fixed a priori.

2. Simple implementation. A typical PROLOG based grammar system will do to realise the scheme presented. In fact, the current version of ANGLABHARTI uses prolog to realise the rule base mentioned. Using the powerful grammar writing system embedded

in QUINTUS-PROLOG [12,32], we have implemented over thirty such rules taking care of the commonly occurring sentence patterns.

Various alternate parsing techniques for natural language grammar may be found in [3, 23].

3. By making a surface structure to surface structure mapping, the tedious scheme of extracting the deep structure, a technique by no means perfected as of now, is eliminated.

4. From psycholinguistics, we infer that the frequency of use of different possible patterns differs. This means that even if all possible patterns are not captured, the common type of sentences can be translated, being limited only by the vocabulary present in the lexicon.

Chapter 3

Logical Design of Lexicon for ANGLABHARTI

3.1 Lexical features

A proper lexicon is a must for any MT activity. For the above system, the following gives a list of salient features required:

- The lexicon should exploit the close relationship among the Indian languages. Some of the features exemplified by the family of Indian languages are:

  1. More often than not, any word in a given sense in a given language has its counterpart in any other language. Hence it is prudent to store in the lexicon, against a word in English in a particular sense, all its counterparts in the target languages. A separate bilingual lexicon for each target language exhibits considerable redundancy in the storage of English components.
  2. Further the lexicon, as preferred, may serve other purposes - MT among Indian Languages is a case in point.

- Each word in English must be disambiguated with regards its senses. A lexicon is a valuable repository to store information for disambiguating word senses.

- In addition to the words in target languages, details such as their morphological information, special grammatical properties may be stored.

- Storage of phrases should be taken into account. It is suggested that phrases be treated as word clusters with regular parts of speech. This enables the parser to treat a single word and a phrase alike, while filling slots for a particular part of speech.

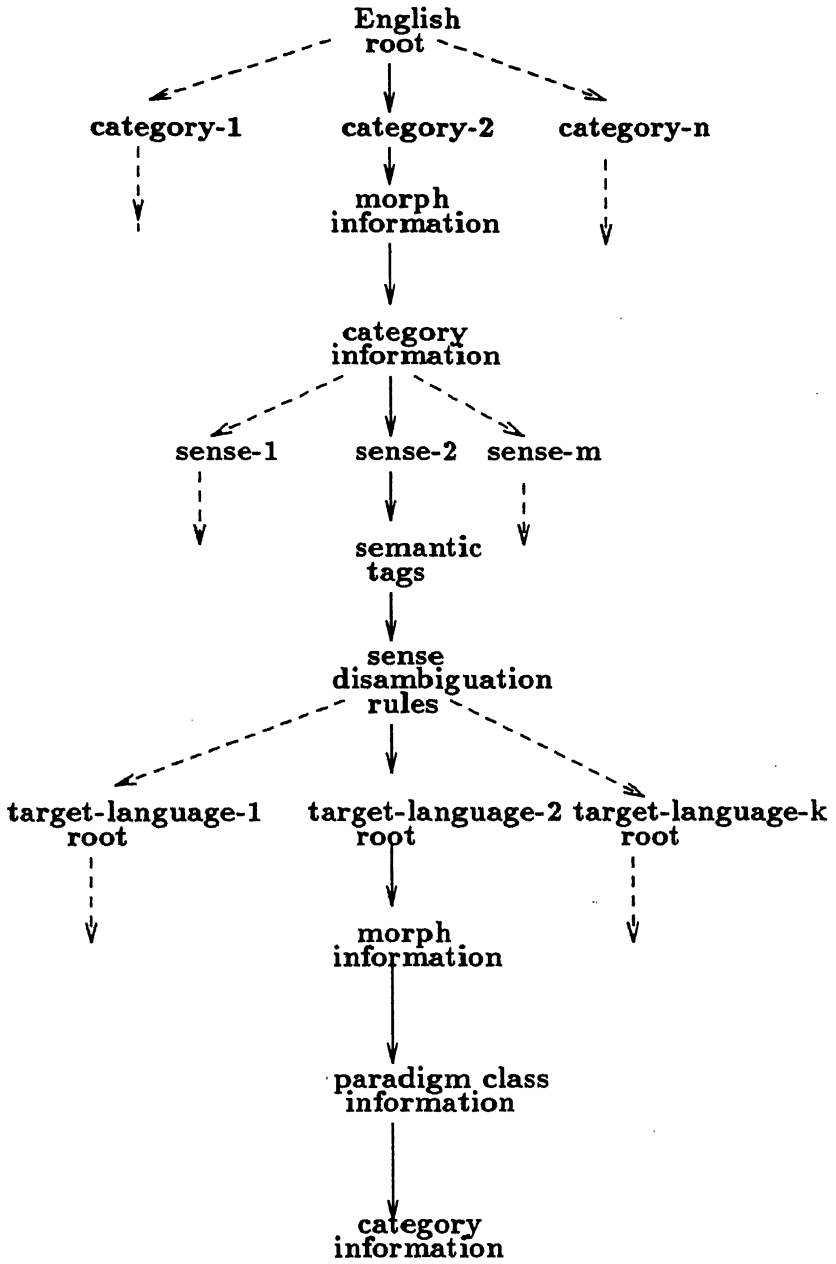


Figure 3.1: Block Schemata of ANGLABHARTI

A special part of speech, ZERO is introduced for phrases, in case they cannot be classified rigidly as any of the regular part of speech. However, a ZERO phrase may be literally substituted in the target language, irrespective of the other units present in the source sentence. Thus ZERO phrases do not affect the regular parse.

- Idioms require special consideration. While a literal substitution of idioms is good enough for MT, it is inefficient. A good lexicon allows the grammatical variations of idioms to be derived by morphological analysis.

3.2 Lexical components

The following components are identified for the lexicon design:

1. The word in English with its grammatical properties. The grammatical properties of a word include its parts of speech and related attributes.

2. Morphology of English word. Details to derive the morphological variations of the root word. For instance 'eat', 'ate', 'eating', 'eaten', 'eats' all may be derived from the root word 'eat'.

3. Semantic tags. These are cryptic semantic primitives for various senses of the word.

4. Sense disambiguation rules. They are the encoded heuristics to identify the proper sense of a word.

5. Roots in target languages, with related information. The gender of a word in English may differ from its counterpart in Hindi, although it is not so in Tamil. Such information must be available for the target text generator. In addition, morphological information for derivations from the roots must be provided.

The lexicon suggested is organised as shown in Figure 3.1.

3.3 Semantic tags

As seen in the diagram, semantic tags are attached to each sense of a word. These tags are strings that provide a common sense classification for the sense involved. These are used by the sense disambiguation rules, to be discussed later.

In order to avoid loading each sense with several possible semantic primitives, a taxonomy of semantic primitives is designed. Thus if the tag reads 'fruit', tags like 'inanimate', 'plant' need not be stored. Therefore lexicon contains nodes in a directed graph of semantic tags, with ancestor nodes being inherited.

The taxonomy construction involves the following stages:

1. Formulation of disambiguation rules. First the heuristics for disambiguating senses of a word is formed. This is done off-line.

2. Directed Acyclic graph construction (DAG). Now the semantic tags are identified and classified corresponding to their real-world relationship.

3. Updating the DAG. As newer semantic tags are identified, they are inserted in the existing DAG.

3.4 Sense disambiguation rules

These are heuristics that enable disambiguation of senses for each word, which are then encoded in the lexicon. These rules make use of grammatical properties (transitivity/intransitivity etc.), patterns (to capture phrasal verbs), semantic tag (to enforce selectional restrictions) and so on.

Elaborate discussion on disambiguation of verbs, prepositions and adjectives can be found in the subsequent chapters.

3.5 off-line and on-line lexicons

In order to take care of storage and efficiency considerations, the lexicon is organised in two parts.

- *off-line lexicon.* This is the data structure containing the bilingual dictionary components for the entire vocabulary used by ANGLABHARTI. Here space is saved by storing all the categories and their related information for each root word together. Currently work is underway to organise it more efficiently as a database.

- *on-line lexicon.* This is the primary data structure that drives ANGLABHARTI. Here organisation is based on category instead of root word. This means that all verbs are

together in a particular format, all nouns are clustered in another format and so on. In particular, if a word can function as a noun as well as a verb, it is repeated.

Another salient feature of on-line lexicon is that the words here appear as they do in the input English sentence. Hence if two morphological derivations of a root word appear in the given sentence, it is stored twice.

The most important feature, of course, is that it caters to the need of a single sentence at hand. As ANGLABHARTI ignores the intersentential context, the system has only information required to handle the current sentence to be translated.

A morphological analyser for English prepares the on-line lexicon from the off-line lexicon. Performance may be improved by preparing the on-line lexicon for a bigger unit like a paragraph, instead of a sentence.

Chapter 4

Ambiguity Resolution in ANGLABHARTI

4.1 Ambiguity

In case there is a single word which tells why MT is difficult, it is *ambiguity*. In communication among humans, several implicit assumptions are available which help us to find the sense of any sentence uttered. However, in MT, elaborate mechanism must be designed to capture this knowledge.

Several cases of ambiguity can be found in Graeme's work [19].

The following gives an idea of how the problem of ambiguity is being tackled by ANGLABHARTI. Whatever ambiguity is not resolved is left for human post-editing.

4.2 Lexical Ambiguity

There are three types of lexical ambiguity:

- *Polysemy.* These are words whose several meanings are related to one another. For eg., the verb 'open' may mean 'unfold', 'expand', 'reveal' etc.
- *Homonymy.* These are words whose meanings are unrelated. For eg., the noun 'bark' may mean 'covering of a tree' or 'noise made by a dog'.
- *Categorical ambiguity.* These are words whose syntactic category can vary. For eg., the word 'sink' may be a noun or a verb.

ANGLABHARTI can tackle both homonymy and categorical ambiguity, as of now. Even typical 'garden path' sentences like "the old dog the footsteps of the young" will be correctly resolved to mean "the footsteps of the young are dogged by the old people".

4.3 Word Sense Disambiguation

The meaning of a sentence and that of its words are closely related by the *Principle of Compositionality*, which insists that the meanings of sentences(and other linguistic expressions consisting of more than one word, such as noun phrases) are understood in terms of the meanings of their component words, and, equivalently, the meanings of words are understood in terms of the contributions they make to the meanings of the sentences in which they occur [27].

The sense in which a word is used in the source sentence should be determined, in order that a suitable word in the target language be chosen. This is so because, as has been observed, different senses of a word maps onto different words in different languages. For example, the noun 'bar' in "the lawyer stopped at the bar for a drink" mean 'a place where drinks are served', rather than 'a court room'. We use a set of syntactic and semantic tags with heuristic rules. This is found to disambiguate a majority of common usage situations.

Typically word sense disambiguation needs:

1. context knowledge. At times this can even mislead. A classic example where context knowledge can actually mislead disambiguation, instead of aiding it, is given by Cherniak [7]. In the sentence, "the astronomer married a star", many people found it difficult to take 'star' for 'a movie star'.

2. local word grouping

3. to handle syntactic disambiguation clues

4. to handle *selectional restrictions* [4]

5. inference, as a last resort

ANGLABHARTI uses 2, 3 and 4.

Selectional restrictions are implemented by the idea of *semantic tags*.

4.4 semantic tags

Semantic tags are keywords that denote the real world usage of a word. For eg., 'programmer' can have tags like 'human', 'skilled-man', 'computer' etc. The idea of semantic tags is pretty old. Aristotle gave nine primitives. Leibnitz used primes to denote concepts. Masterman's

semantic net [28] uses 100 primitives, while Schank [35] uses 11 primitives. A similar concept called *semantic formulae* was also advocated by Wilks [42].

The idea of using semantic tags to link a word and the object it denotes is closely related to the the concept of *reference* in linguistics [22]: "*the relationship between word and object is called the relationship of reference*". In fact, in linguistics, the meaning of a word is attempted to be equated with the relationship of reference (*extensionalism*).

The following gives a partial list of semantic tags, taken from Beaugrande [5]:

1. *object*: conceptual entities with a stable identity and constitution
2. *situation*: configurations of mutually present objects in their current states
3. *event*: occurrences which change a situation or a state within a situation
4. *action*: events intentionally brought about by an agent
5. *state*: the temporary, rather than characteristic, condition of an entity
6. *relation*: a residual category for incidental, detailed relationships like 'father-child', 'boss-employee', etc.
7. *location*: spatial position of an entity
8. *time*: temporal position of a situation, state or event
9. *motion*: change of location
10. *instrument*: a non-intentional object providing the means for an event
11. *form*: shape, contour, and the like
12. *substance*: materials from which an entity is composed
13. *containment*: the location of one entity inside another but not as a part or substance
14. *perception*: operations through sensory organs
15. *cognition*: storing, organizing, and using knowledge by sensorially endowed entity
16. *emotion*: an experientially or evaluatively non-neutral state of a sensorially endowed entity

17. *volition*: activity of will or desire by a sensorially endowed entity

18. *recognition*: successful match between perception and prior cognition

19. *communication*: activity of expressing and transmitting cognitions by a sensorially endowed entity

20. *possession*: relationship in which a sensorially endowed entity is believed (or believes itself) to own and control an entity

21. *quantity*: a concept of number, extent, scale, or measurement

22. *value*: assignment of the worth of an entity in terms of other entities

Most of these concept types are familiar from *case grammar* that undertook to classify language relationships according to the organization of events and situations [15, 16].

The troubles occurring with the usage of semantic tags are: [36]

- A universal set of primitives is yet to be identified.
- Finer shades of meaning among a family of synonyms can cause trouble.
- Grain of classification depends on application.

Yet the approach is simple and effective, when the lexicon is carefully designed.

4.5 Ambiguity Preservation

A crucial trick used by ANGLABHARTI is to transfer the ambiguity over the target languages, whenever that is possible. For example, consider the sentence [24]
"the quarrelsome Arabs want another war"
This may be translated into Tamil as
"kObakkAra arAb_kArargal matrum oru sandai kEtkirArgal".
In both cases, there is an ambiguity, rending two readings:

1. The Arabs want another war and they are quarrelsome

2. Only those Arabs who are quarrelsome wish to fight again

Clearly the decision whether to disambiguate or not in such cases should be present in the appropriate pattern in rule base.

Subsequent chapters discuss how ambiguity resolution is carried out in specific cases.

Chapter 5

On disambiguating verb senses in English

5.1 Multiple verb senses

Typically, any verb encountered in English has more than one sense. In any NLP system, it becomes essential to find the correct sense of usage, depending on the sentence and its context. During MT, different senses of a verb can have different equivalents in the target language. For instance, the English verb 'bark' as in "dogs bark" is equivalent to 'kuraithal' in Tamil, whereas the same verb as in "he barked his shins against some stone steps" is equivalent to 'sirAithukk_kolludal' in Tamil.

The present chapter deals with resolving the multiple senses of a verb in a sentence by simple techniques like checking its syntactic attributes, eg. whether it is transitive/intransitive, and by observing the semantic tags attached to the nouns surrounding it.

5.2 Disambiguating with syntactic attributes

One can often find the sense of a verb by observing the syntactic usage of each of its senses. Consider the verb 'navigate' which has the following senses:-

1. Intransitive verb. find the position and plot the course of a ship, an aircraft, a car, etc. using maps and instruments.
   Which officer in the ship navigates?
   I'll drive the car: you navigate, i.e. tell me which way to go.

2. Transitive-verb - noun / Transitive-verb - noun - prepositional-phrase:
   steer(a ship); pilot an aircraft;

Navigate the tanker round the Cape.

(figurative) Navigate a Bill through Parliament

Any good dictionary gives valuable clues about the syntactic usages of different senses of a verb, which can be immediately harnessed.

5.3 Disambiguating with verb patterns

Many usages of verb has a fixed pattern. By recognising this pattern, the sense can be decided. This is especially true of idioms and phrasal verbs.

This is illustrated with the usage of verb 'put': (sb: somebody sth: something)

- put sth about:

  spread or circulate (false news, rumours, etc.)

  eg. he's always putting about malicious rumours.

  it's being put about that the Prime Minister may resign.

- put sth across sb:

  trick sb into accepting a claim, etc that is worthless or untrue.

  eg. Are you trying to put one across me?

- put oneself/sth across/over (to sb):

  communicate or convey (one's personality, an idea, etc) to sb

  eg. He doesn't know how to put himself across at interviews.

  she's very good at putting her ideas across.

- put sth at sth:

  calculate or estimate (the size, cost, etc of sth) to be (the specified weight, amount, etc)

  eg. I would put his age at about sixty.

  what would you put the price of this car at?

  I'd put it at $15000.

- put sb away:

  confine sb in a prison or mental hospital

  eg. He was put away for ten years for armed robbery.

  She went a bit odd and had to be put away.

It should be of interest to note that many of the senses can be resolved by such pattern analysis.

5.4 Disambiguating with voices of the sentence

When the above features fail to help, the voices of the sentence may help to disambiguate the sense involved.

For eg. consider the 'transitive-verb - noun' usage of the verb 'run'.
in active voice: (among many other senses) cover (the specified distance) by running eg. Who was the first man to run a mile in under four minutes?
in passive voice: cause a race to take place.
eg. The Grand National will be run in spite of the bad weather.

5.5 Disambiguating with semantic tags

This is a much more powerful technique than the ones discussed. However, this is much fuzzier also. This requires an in-depth analysis of the usage of a verb sense before identifying the disambiguating rules.

In this method, we identify the nature of the subjects and objects involved with their real-world usage. Typically cryptic semantic primitives are kept in the lexicon with the nouns. Using these semantic primitives, the sense of a verb can be resolved.

Consider the different senses of the verb 'fall':

1. come or go down from force of weight, loss of balance, etc.; descend or drop.
   eg. The rain was falling steadily
   The leaves fall in autumn
   He slipped and fell ten feet
   That parcel contains glass - don't let it fall
   The book fell off the table onto the floor
   He fell into the river
   I need a new bicycle lamp - my old one fell off and broke
   RULE: Subject is a physical-object

2. hang down
   eg. Her hair fell over her shoulders in a mass of curls

His beard fell to his chest

RULE: Subject is a **physical-object, fixed-at-one-end**

3. decrease in number, amount or intensity

   eg. Prices fell on the stock market

   Her spirits fell at the bad news

   Her voice fell as they entered the room

   The temperature fell sharply in the night

   RULE: Subject: **quantifiable.**

4. lose one's power, office or position; be defeated

   eg. The Government fell after the revolution

   RULE: Subject: **officer-bearer**

5. die in battle; be shot

   eg. Half the regiment fell before the enemy onslaught

   Six tigers fell to his rifle

   RULE: Prepositional-phrase: **gun** and Subject: **animate**

6. (of a fortress, city etc) be captured

   eg. Troy finally fell to the Greeks

   RULE: Subject: PLACE

7. happen or occur; has as a date

   eg. Easter falls early this year

   Christmas day falls on Monday

   RULE: Subject: **event**

8. be spoken

   eg. Not a word fell from his lips

   RULE: Subject: **speech**

Thus this method essentially involves a linguist's conclusion about the nature of the phrases involved in the sentence. The grain of semantic tags required clearly depends on how close two related senses are.

5.6 Defects of sense disambiguator in ANGLABHARTI

1. By and large, this is an ad-hoc scheme. Whenever the rules for disambiguation are found to be inadequate, they must be revised.

2. Being a shallow approach, this may not be used for a rigorous understanding of the senses. There may be cases where the senses cannot be disambiguated by merely the above techniques.

3. This involves considerable *Human Engineering*. All the rules for disambiguation must be manually identified and put in the lexicon.

4. Considerable skill is required for the identification of semantic tags. A brute force method of assigning semantic tags can cause enormous growth in the number of tags to be produced.

5.7 Merits of sense disambiguator in ANGLABHARTI

1. Being a simple technique, this is a readily computationally feasible approach. Especially this is true when the domain of text is known *a priori*.

2. Data for disambiguation is readily available from a good dictionary.

3. Due its sole dependence on syntactic entities like noun-phrases and prepositional-phrases, the disambiguator module can readily use the parser output.

4. ANGLABHARTI envisages post-editing. Therefore the method suggests itself provided the alternatives generated are not very high.

Chapter 6

Disambiguating Adjectives using Selectional Restrictions

6.1 Word Sense ambiguity of Adjectives

Typically adjectives have multiple senses, usually determined by the context of their usage. For instance, the adjective 'green' has at least 8 different senses. In any MT system, before the proper translation can be generated, it is essential to find the appropriate sense of an adjective present in the sentence. ANGLABHARTI tackles the situation in the sense disambiguator module, in the same vein as it does for verbs. Essentially, by observing the other adjectives or noun that follows, using the semantic tags attached to them, an attempt is made to find out the proper sense.

Note that an adjective is related with other adjectives and the noun that it modifies by rule base. As an example, "noun is adj" pattern connects the adjective with the subject. It is true that not always such a relationship is uniquely determined by ANGLABHARTI. Where required, multiple associations of adjectives are permitted. A human engineered post-editor is assigned the task of chosing the proper output.

As in the case of verbs, disambiguation of adjectives requires a proper analysis of their usages, which is then encoded in the lexicon.

6.2 Selectional Restrictions of Adjectives

While essentially the same approach used for disambiguating senses of any word in general (viz. using patterns, syntactic characteristics etc. **[Section 4.3]**) can be applied here, ANGLABHARTI uses only the semantic tags in this process. This is because, it is believed that, they are

the tools most useful in this case. Future versions of ANGLABHARTI may use other approaches as well.

The following gives an illustration of the approach used:

- *competent*

  1. having the necessary ability, authority, skill, knowledge, etc.
     eg. a highly competent driver; he is not competent to look after young children
     noun: *human*
  2. quite good but not excellent
     eg. the novel may be a best seller, but it's no more than a competent piece of writing;
     a competent piece of work
     noun: *work*

- *green*

  1. of the color between blue and yellow in the spectrum
     eg. fresh green peas
     noun: *physical-object*
  2. covered with grass or other plants
     eg. green fields, hills, etc.
     noun: *geological-object*
  3. not yet ripe
     eg. green bananas; apples too green to eat
     noun: *fruit*
  4. not yet dry enough for use
     eg. green wood does not burn well
     noun: *wood*
  5. immature; inexperienced; easily fooled
     eg. a green young novice; you must be green to believe that
     noun: *human*

- *plain*

1. easy to see, hear or understand; clear

   eg. the markings along the route are quite plain; in plain English; He made it plain to us that he did not wish to continue; she made her annoyance plain.

   noun: *abstract-idea* or *communication*

2. not decorated or luxurious; ordinary and simple

   eg. a plain but very elegant dress; a plain food/cooking; plain cake

   noun: *product*

3. not beautiful or good looking

   eg. a few rather plain bits of furniture

   from a rather plain child, she had grown into a beautiful woman

   noun: *human* or *physical-object*

- *replete*

1. well-fed or full; gorged

   eg. lions replete with their kill; feel replete after a large meal

   noun: *human* or *animal*

2. well stocked or supplied

   eg. a house replete with every modern convenience

   noun: *default*

- *serious*

1. solemn and thoughtful; not frivolous

   eg. a serious person, mind, appearance

   her face was serious as she told us the bad news

   he seems very serious, but in fact he has a delightful sense of humour

   please be serious for a minute, this is very important

   noun: *default*

2. intended to provoke thought; not merely for amusement

   eg. a serious essay about social problems

   do you ever read serious works?

   noun: *work*

3. important because of possible danger or risk; grave
   eg. a serious illness, mistake, accident
   a serious decision about giving up a steady job
   that could cause serious injury
   the international situation is extremely serious
   noun: *unfavourable*

- *volatile*

  1. changing rapidly into vapour
     noun: *liquid*
  2. changing quickly from one mood or interest to another; fickle
     eg. a slightly volatile personality, disposition, nature, etc. noun: *person* or *personality*
  3. likely to change suddenly or sharply; unstable
     eg. volatile stock-markets, exchange rates; a volatile political situation
     noun: *trade* or *state-of-affair*

6.3 Demerits of the Scheme

The demerits of using semantic tags can be easily identified:

1. *ad-hoc scheme.* By and large, this is an ad-hoc scheme, as no universal set of semantic primitives are yet identified.
2. *sufficiency.* Clearly this is insufficient. A more general scheme involving other properties mentioned earlier is necessary to disambiguate all cases.
3. *difficult lexicon development.* Clearly the lexicon should be built carefully after considering all the cases. However, with the help of a good lexicon, this obstacle can be minimised.

A detailed analysis on the defects on relying on semantic tags alone for disambiguation and the various remedies thereof can be found in Tennant[37].

6.4 Merits of the Scheme

1. *simplicity.* The scheme is basically simple to implement. Surprisingly this seems to take care of majority of the cases in daily usage.

2. *no extra effort required.* Semantic tags are already present in the lexicon for each noun, as this is required by the verb disambiguator. By using the existing information in the lexicon for disambiguating adjectives, considerable effort is saved.

Chapter 7

Mapping Prepositions from English to Tamil

7.1 Preposition Mapping

During Machine Translation from English to Tamil, one ends up finding the equivalent of the English prepositions in Tamil. The usual techniques involve a deep analysis of the sentence, to resolve the case roles played by the various phrases involved. However, a simpler scheme is suggested which is independent of other components in the sentence with a prepositional phrase.

This chapter deals with prepositions of the form 'noun-phrase preposition noun-phrase'. The suggestion is that by simply knowing the semantic tags attached to the head nouns on either side, one can map the prepositions from English to Tamil.

It should be noted that English is a verb-central language while Tamil is a verb-final language. Hence the movement of prepositional phrases from English to Tamil is: preposition noun-phrase $\rightarrow$ noun-phrase' preposition'.

7.2 Mapping

The mapping of prepositions is demonstrated henceforth. The following heuristics are arranged in the descending order of priority, for each preposition considered. 'anything' is a wild card, while 'nil' refers to absence of a noun-phrase, which is being modified by the preposition considered. Note that in the latter case, the mapping is bound to be erroneous in some cases, as the classification of the verb preceding the preposition is neglected, to simplify the matter.

It is of interest to note that this arrangement enables invoking a default mapping for any preposition.

- **on**

  human *on* physical-body → -il_ulla

  eg. the people on the bus → bus -il_ulla makkal

  animal *on* physical-body → -il_ulla

  eg. the dog on the terrace → mAdi -il_ulla nAi

  nil *on* physical-body → -il

  eg. the people were singing on the bus → makkal bus -il pAdikkondirundArgal

  he sat on the chair → avan nArkAli -il utkArndAn

  nil *on* time → -andru

  eg. on Tuesday, I came → Tuesday -andru nAn vandEn

  event *on* anything → -ai_patri

  eg. the course on English grammar → English ilakkanam -ai_patri pAdam

- **in**

  nil *in* anything → -il

  eg. in the afternoon, we went to Boston → mAlai -il (we went to Boston)'

  the load arrived in March → sarakku March -il vandu_irangiyadu

  in a grave manner → (grave manner)' -il

- **to**

  nil *to* place → -ikku

  eg. we went to his house → nAngal avan vIdu -ikku pOnOm

  my house is next to yours → enadu vIdu ungaludaiyadu -ikku aduttu ulladu

  (Note that semantic tag for 'yours' is borrowed from that for the head noun of the subject 'my house')

- **at**

  nil *at* physical-body → -ai

  eg. we were looking at his awful paintings →

  nAngal avanadu mOsamAna Oviyangal -ai pArthukkondirundOm

  nil *at* human → -mIdu

  eg. he was surprised at her → avan aval -mIdu AcharyappattAn

  nil *at* action → -mIdu

eg. the public were shocked at his murder → avanadu kolai -mIdu makkal vagundArgal

- **near**

  nil *near* anything → -arugE

  eg. the man went near the club → manidan club -arugE chenrAn

  human *near* anything → -arugEa_ulla

  eg. the man near the table → table -arugE_ulla manidan

- **for**

  anything *for* time → -ikku

  eg. he is expected for several weeks → avan pala vArangal -ikku ethirpArkkappadugirAn

  their arrival for a month → oru mAsam -ikku avargaladu varugai

  anything *for* anything → -ikkAga

  eg. I worked for success → nAn vatri -ikkAga uzhaikkirEn

  her friendship for Chopin → Chopin -ikkAga avaladu natpu

- **during**

  nil *during* time → -pozhudu eg. they worked during the vacation → avargal llvu_nAtkal -pozhudu vElai_saidArgal

- **of**

  anything *of* abstract-idea → -ai_patri

  eg. he convinced her of the need → avan avallukku tEvai -ai_patri puriya_vaithAn

  (Note: 'of the need' does not modify 'her')

  the news of her success → avaladu vetri -ai_patri saidi

  event *of* human → -udaiya

  eg. the arrival of his daughter → avanadu magal -udaiya varugai

  an opera of Verdi's → Verdi -udaiya oru opera

  place *of* place → nil

  eg. the city of Rome → Rome nagaram

  anything *of* anything → -in

  eg. guard of the house → vIdu -in kAval_kAran

the king of Spain → Spain -in arasan

the day of her arrival → avaladu varugai -in dinam

the enthusiastic reception of the play → nAdagam -in urchAga varavErpu

- **before**

  time *before* event → -ikku_mudal

  eg. the day before her arrival → avaladu varugi -ikku_mudal dinam

- **between**

  anything *between* anything → -ikku_idaiyil

  eg. she came between 2'O clock and 3'O clock →

  aval 2 mani matrum 3 mani -ikku_idaiyil vandAl

- **from**

  event *from* place → -il_irundu

  eg. the departure from Hamburg → Hamburg -il_irundu purappAdu

- **behind**

  nil *behind* physical-body → -in_pinnE

  eg. behind the bus → bus -in_pinnE

  anything *behind* physical-body → -in_pinnE_ulla

  eg. the children behind the fence → vEli -in_pinnE_ulla

- **over**

  human *over* physical-body → -mIdu_ulla

  eg. the man over the bridge → bridge -mIdu_ulla manidan

  anything *over* anything → -mIdu

  eg. the quarrel over pay → sambalam -mIdu sandai

- **by**

  nil *by* anything → -Al

  eg. they were welcomed by the hosts → avargal hosts -Al varavErkkappattanar

  anything *by* anything → -in

  eg. work by the artists → kalaigyargal -in vElai

It is hoped that further addition of such transformation rules can improve the quality of Tamil text generator in ANGLABHARTI.

Chapter 8

A Template-Driven Morphological Derivation for Verbs in Tamil

8.1 Verb Variations

ANGLABHARTI uses a rule base which converts sentences from English into an intermediate form, which is then fed to the text generator modules for different target language. The text generator routine derives the appropriate form of the target word from the root word and the other information provided.

The present chapter deals with deriving the variations of a verb in Tamil from its root, which can be used by a text generator as above for Tamil. The paradigm envisaged here is to use the idea of *verb-templates* and *verb-classes*. A verb-template is the part pertaining to verbs in the intermediate form generated from the rule base. For example, [has-already-been-verb_ing-form] is a template. A verb class refers to a group of verb roots in Tamil, which behave in the same way during morphological derivation. For example, 'vaithiru' and 'iru' are two such verb roots in Tamil that have similar suffixes during variations under tense, gender, number, person etc.

By constructing rules for a given template and a given class and by classifying the existing verbs in Tamil into different classes, one gets a neat scheme to derive the variations of verb, which can be used for a text generator as discussed earlier.

The idea is elaborated in the subsequent sections.

8.2 Verb Templates

Five morphological variations of a verb in English is observed. Labeling them suitably, we have

verb-1: eat form (eat, walk, come, run etc.)

verb-2: eats form (eats, walks, comes, runs etc.)

verb-3: ate form (ate, walked, came, ran etc.)

verb-4: eaten form (eaten, walked, come, run etc.)

verb-5: eating form (eating, walking, coming, running etc)

We have modal-words like 'must', 'can', 'may be' etc. In addition, there are auxiliaries like 'am', 'is', 'were' etc. Infinitives are preceded by the word 'to'. Negatives are typically denoted by the word 'not'.

Armed with this classification, one can easily identify the different templates for the occurrence of a verb phrase in English. The following gives a sample:

verb-3(): walked, worked, helped

verb-1(): walk, work, help

am-verb-5(): am walking, am working, am helping

does-verb-1(): does walk, does work, does help

is-already-verb-5(): is already walking, is already working, is already helping

hope-to-verb-1(): hope to walk, hope to work, hope to help

might-have-verb-4(): might have walked, might have worked, might have helped

should-not-verb-1(): should not walk, should not work, should not help

will-not-verb-1(): will not walk, will not work, will not help

was-verb-3(): was helped

Now such a template with the gender, number, person of the subject and the root verb in the target language provides all the details necessary to find the appropriate verb that should appear in the output stream of ANGLABHARTI.

It should be noted that in addition, minor changes may be required. For example, when the verb-phrase is used in an imperative sentence as in "go to school", the gender, number and person parameters hold no significance. This detail must be conveyed to the module that derives the target verb.

In ANGLABHARTI, about 50 such templates of verb-phrases are identified.

8.3 Verb Classes

As already mentioned, these are groups of roots that behave identically during morphological derivation. As many as 30 such classes are identified for Tamil. The lexicon contains details

about the class of each verb. It should be noted that for a given verb class, there are as many rules for deriving suffixes as there are combinations of templates and gender, number and person of subject.

The current version of ANGLABHARTI merely adds a suitable suffix to the verb-root. The future versions are expected to perform *sandhi analysis*, i.e. analysis of how the final letters of the verb-root get modified, while adding a suitable suffix.

8.4 Demerits of the Scheme

1. Too many (template - gender, number, person) combinations possible.

2. Redundancy in templates. For instance, *may-verb-1()* form and *can-verb-1()* form both behave similarly for Tamil, with respect to morphological derivation. eg. "may go" and "can go" both corresponds to 'pOgalAm'.

8.5 Merits of the Scheme

1. *Minimal analysis of verb phrases.* This obviates the need to extract the tense and modality information.

2. *Uniform approach to all target languages.* A similar approach is used in the text generators of Hindi and Telugu.

Chapter 9

Conclusion

9.1 Current Implementation

The current version of ANGLABHARTI uses Prolog platform to realise the rule base, the Sense Disambiguator and the Tamil Text Generator. However other activities like the human-engineering aspect, the preparation of *on-line lexicon*, Text Generators for Hindi and Telugu, etc. are done in the *C* Programming Language. Within a period of over a year, a workable version of ANGLABHARTI is available.

9.2 Future Activities

The following are some of the future activities envisaged:

1. *Lexicon Development.* A massive multi-lingual vocabulary involving English, Tamil, Hindi and Telugu is envisaged.

2. *Rule Base Augmentation.* The rule base is to be augmented to cover more number of structures.

3. *Text Generation.* The Text Generators for the different target languages is to be enlarged, by increasing the verb-classes.

4. *More Disambiguation.* More number of categories is to be disambiguated, based on the principles already being used.

5. *Human Engineering.* ANGLABHARTI is to be made more user-friendly.

Appendix A

Sample Rule Base in ANGLABHARTI

s_body(Op) →

noun_phrase(G, N, P, _, Tag1, A),

verb_phrase(normal, G, N, P, Voice, Vrules, B), prep_phrase(['NIL'], _, Ptag, C),

{ Srules=['Ipr'], resolve(Vrules, Srules, Ptag, Voice, Tag1, ['NIL'], ['NIL']),

append(A, C, T1), append(T1, B, Op) }

% I am going to the market

% I was attacked by him

% I was attacked by night

| noun_phrase(G, N, P, _, Tag1, A), verb_phrase(normal, G, N, P, Voice, Vrules, B),

prep_phrase(['NIL'], Ntag, Ptag, C), noun_phrase(_, _, _, _, Tag2, D),

{ Srules=['Dnpr'], resolve(Vrules, Srules, Ptag, Voice, Tag1, Tag2, Ntag), append(A, D, T1),

append(T1, C, T2), append(T2, B, Op) }

% I gave to the boy a toy

% the boy was given by me a toy

| noun_phrase(_, singular, third, _, _, A), [is], noun_phrase(_, _, _, _, _, B),

{ !, append(A, B, Op) }

% she is my sister

% we dread Mary/Mary's taking over the business

| [there, is, something], adv(B), [about], noun_phrase(_, _, _, _, _, C),

{ append(C, [l1], T1), append(T1, [l9], T2), append(T2, B, T3), append(T3, [v1], Op) }

% there is something pleasing about him

```
| question(A), [is], noun_phrase(G, singular, third, _, _, B),
{ is_verb(G, V), append(B, A, T1), append(T1, V, Op) }
% where is the girl
and so on
option_1 → [that] | [ ].
rest_same(X1, Op) → [and], sentence(more, X), { append(X1, [13], T1), append(T1, X, Op) }.
rest_rev(X1, Op) →
[after], sentence(more, X), { append(X, [17], T1), append(T1, X1, Op) }
| [when], sentence(more, X), { append(X, [18], T1), append(T1, X1, Op) }
| [because], sentence(more, X), { append(X, [12], T1), append(T1, X1, Op) }
| [since], sentence(more, X), { append(X, [15], T1), append(T1, X1, Op) }.
sentence(Level, Op) → s_body(Op1), rest(Level, Op1, Op).
rest(Level, X, Op) →
[ ], { Level=more, Op=X }
| [W], { lastword(W), Level=one, append(X, [W], Op) }
| rest_same(X, X1), rest(Level, X1, Op)
| rest_rev(X, X1), rest(Level, X1, Op).
```

Appendix B

Sample Rule Base for Nouns in ANGLABHARTI

```
noun_body(G, N, P, Tag, X) →
pos_case(A), det_star(B), noun(G, N, P, Tag, C), { append(A, B, T1), append(T1, C, X), ! }
% his two bags
| verb_5(noun, normal, G, N, P, _, A), adv(B), { append(B, A, X) }
% eating greedily
| verb_5(noun, normal, G, N, P, _, X)
| adv(A), verb_5(noun, normal, G, N, P, _, B), { append(A, B, X) }
| det_star(A), adj_star(B), noun(G, N, P, Tag, C), { append(A, B, T1), append(T1, C, X) }.
% the great man
rest_n_same(V1, G1, N1, P1, G, N, P, Op) →
[whose], sentence(more, A),
{ append(['enda'], V1, T1), append(T1, ['-udaiya'], T2),
append(T2, A, T3), append(T3, ['-O'], T4),
append(T4, ['anda'], T5), append(T5, V1, Op), G=G1, N=N1, P=P1 }
% enda noun -udaiya sentence -O anda noun
| [which], s_minus_np1(A),
{ G1=neuter, append(A, V1, Op), G=G1, N=N1, P=P1 }
% s_minus_np1 NP
| [', '], { V1=Op, G=G1, N=N1, P=P1 }
| [who], s_minus_np1(A),
{ not(G1=neuter), append(A, V1, Op), G=G1, N=N1, P=P1 }
```

```
% s_minus_np1 NP
| [and], noun_phrase(_, _, _, _, _, A), { append(V1, [13], T1), append(T1, A, Op),
G=dont_care, N=plural, P=third }
| [', '], noun_phrase(_, _, _, _, _, A), { append(V1, [', '], T1), append(T1, A, Op), G=dont_care,
N=plural, P=third }
| [or], noun_phrase(G, N, P, _, _, A), { append(V1, [14], T1), append(T1, A, Op) }.
rest_n_rev(Tag, A_tag, A_tag1, V1, Op) →
prep_phrase(Tag, _, Ptag, X), { append(X, V1, Op), append(A_tag, Ptag, A_tag1) }
| verb_5(non_finite, normal, _, _, _, _, A), prep_phrase(['NIL'], _, _, B),
{ A_tag1=A_tag, append(B, A, T1), append(T1, V1, Op) }.
noun_phrase(G, N, P, All_tag, Tag, Op) →
noun_body(G1, N1, P1, Tag, X1), rest_n(Tag, Tag, All_tag, X1, G1, N1, P1, G, N, P, Op).
rest_n(Tag, A_tag, All_tag, V1, G1, N1, P1, G, N, P, Op) →
[ ], { G=G1, N=N1, P=P1, Op=V1, All_tag=A_tag }
| rest_n_same(V1, G1, N1, P1, G2, N2, P2, X1), rest_n(Tag, A_tag, All_tag, X1, G2, N2, P2,
G, N, P, Op)
| rest_n_rev(Tag, A_tag, A_tag1, V1, X1), rest_n(Tag, A_tag1, All_tag, X1, G1, N1, P1, G, N,
P, Op).
```

Appendix C

Sample Rule Base for Verbs in ANGLABHARTI

v_body(Type, G, N, P, Voice, Vrules, X) →

verb_3(Type, normal, G, N, P, Vrules, X), { Voice='active' }

| verb_1(Type, normal, G, N, P, Vrules, X) , { Voice='active' }

| verb_2(Type, normal, G, N, P, Vrules, X), { Voice='active' }

| [am], verb_5(Type, am, G, N, P, Vrules, X) , { Voice='active' }

| [can], verb_1(Type, can, G, N, P, Vrules, X), { Voice='active' }

| [had, been], verb_5(Type, had_been, G, N, P, Vrules, X) , {Voice='active' }

| [hope, to], verb_1(Type, hope_to, G, N, P, Vrules, X), { Voice='active' }

| [is, going, to], verb_1(Type, is_going_to, G, N, P, Vrules, X) , { Voice='active' }

| [might, have], verb_4(Type, might_have, G, N, P, Vrules, X) , { Voice='active' }

rest_v_same(Type, V1, G, N, P, Op) → [and], verb_phrase(Type, G, N, P, _, _, A), { append(V1, l3, T1), append(T1, A, Op) }.

rest_v_rev(_, V1, _, _, _, Op) → adv(X), { append(X, V1, Op) }.

v_prefix(X) → [], { X=[] }

| adv(X).

verb_phrase(Type, G, N, P, Voice, Vrules, Op) → v_prefix(X), v_body(Type, G, N, P, Voice, Vrules, X1),

{ append(X, X1, X2) }, rest_v(Type, X2, G, N, P, Op).

rest_v(Type, V1, G, N, P, Op) → [], { Op=V1}

| rest_v_same(Type, V1, G, N, P, Op)

| rest_v_rev(_, V1, _, _, _, Op).

Appendix D

Suffixes for Verb Roots in Tamil

vsuffix('adai',one,normal,normal,dont_care,singular,first,['kkirEn']).

vsuffix('pidi',one,normal,can,dont_care,singular,first,['kka_vallavan']).

vsuffix('pidi',three,normal,normal,neuter,singular,third,['thadu']).

vsuffix('pidi',three,normal,normal,dont_care,plural,third,['ttArgal']).

vsuffix('sAppidu',one,normal,normal,neuter,plural,third,['girana']).

vsuffix('sAppidu',one,command,never,dont_care,dont_care,dont_care,['Ade']).

vsuffix('sAppidu',two,non_finite,normal,dont_care,dont_care,dont_care,['ginra']).

vsuffix('sAppidu',five,noun,normal,neuter,singular,third,['vadu']).

vsuffix('sAppidu',three,normal,normal,dont_care,singular,second,['ittAi']).

vsuffix('pO',one,normal,normal,dont_care,plural,third,['girArgal']).

vsuffix('pO',three,normal,normal,dont_care,plural,third,['nArgal']).

vsuffix('pO',five,normal,am,dont_care,singular,first,['girEn']).

vsuffix('pO',five,normal,is,neuter,singular,third,['giradu']).

vsuffix('iru',one,normal,normal,feminine,singular,third,['kkirAl']).

vsuffix('iru',one,normal,normal,masculine,singular,third,['kkirAn']).

vsuffix('iru',one,normal,normal,neuter,singular,third,['kkiradu']).

vsuffix('iru',two,normal,normal,neuter,singular,third,['kkiradu']).

vsuffix('iru',three,normal,normal,feminine,singular,third,['ndAl']).

vsuffix('iru',three,normal,normal,masculine,singular,third,['ndAn']).

vsuffix('pAr',one,non_finite,normal,dont_care,dont_care,dont_care,['kka']).

vsuffix('pAr',two,non_finite,normal,dont_care,dont_care,dont_care,['kkinra']).

vsuffix('pAr',three,normal,normal,dont_care,singular,first,['thEn']).

vsuffix('pAr',four,non_finite,have_been,dont_care,dont_care,dont_care,['kkappattadAga']).

vsuffix('vAzh',one,non_finite,normal,dont_care,dont_care,dont_care,['a']).

vsuffix('pEsu',two,normal,normal,masculine,singular,third,['girAn']).

vsuffix('vA',three,normal,normal,dont_care,singular,first,['andEn']).

vsuffix('utkAr',three,normal,normal,masculine,singular,third,['ndAn']).

vsuffix('sol',three,normal,normal,masculine,singular,third,['nnAn']).

vsuffix('sol',three,normal,normal,dont_care,singular,first,['nnEn']).

vsuffix('anuppu',three,normal,were,dont_care,plural,third,['ap_pattanar']).

vsuffix('anuppu',three,normal,was,masculine,singular,third,['ap_pattAn']).

vsuffix('anuppu',three,normal,was,feminine,singular,third,['ap_pattAl']).

vsuffix('anuppu',three,normal,was,neuter,singular,third,['ap_pattadu']).

vsuffix('anuppu',three,normal,normal,masculine,singular,third,['inAn']).

vsuffix('anuppu',three,normal,normal,neuter,singular,third,['iyadu']).

vsuffix('vA',three,normal,normal,dont_care,singular,second,['andAi']).

vsuffix('kodu',three,normal,normal,feminine,singular,third,['thAl']).

vsuffix('kEl',three,normal,normal,feminine,singular,third,['ttAl']).

vsuffix('Odu',three,normal,normal,feminine,singular,third,['inAl']).

vsuffix('Odu',three,normal,normal,masculine,singular,third,['inAn']).

vsuffix('sai',three,normal,normal,feminine,singular,third,['dAl']).

vsuffix('Agu',three,non_finite,normal,dont_care,dont_care,dont_care,['iya']).

vsuffix('kurai',three,normal,normal,neuter,singular,third,['thadu']).

vsuffix('kurai',three,normal,normal,masculine,singular,third,['thAn']).

vsuffix('kurai',three,normal,normal,feminine,singular,third,['thAl']).

vsuffix('kathu',three,normal,normal,masculine,singular,third,['inAn']).

vsuffix('thirudu',four,normal,was,neuter,singular,third,['appattadu']).

vsuffix('nil',five,non_finite,normal,dont_care,dont_care,dont_care,['irkinra']).

vsuffix('thiri',three, normal,normal,dont_care,plural,third,['ndanar']).

vsuffix('thiri',three, normal,normal,neuter,singular,third,['ndadu']).

vsuffix('thiri',three, normal,normal,masculine,singular,third,['ndAn']).

vsuffix('thiri',three, normal,normal,feminine,singular,third,['ndAl']).

Format: vsuffix(root, type1, type2, type3, gender, person, suffix)

type1 = one | two | three | four | five

Appendix E

Bibliography

[1] Akin P., Conwell M. J., *Patterns in Language and Writing - An Integrated Approach*, D. Van Nosrand Company, 1979

[2] Allen J., *Semantic Interpretation Strategies*, in *The Handbook of Artificial Intelligence - Volume IV*, ed. A.Barr, P.R.Cohen and E.A.Feigenbaum, Addison-Wesley Publishing Company, Inc., 1989, pp. 213-222

[3] Allen J., *Natural Language Understanding*, The Benjamin/Cummings Publishing Company, Inc., 1987

[4] Allen J. , Van Buren P. (eds), *Chomsky: Selected Readings*, Chapter: 5, Oxford University Press. 1971,

[5] Beaugrande R. D., *Text, Discourse, and Process*, Norwood, N.J., 1980

[6] Berwick R.C., *The Acquisition of Syntactic Knowledge*, The MIT Press, 1985

[7] Cherniak, Cognitive Science 7, 1983, pp. 171-190

[8] Chomsky N., *Three Models for the Description of Language*, IRE Trans. Inform. Theory, IT-2, No. 3, 1956, pp. 113-124

[9] Chomsky N., *Aspects of the Theory of Syntax*, Cambridge, Mass.: The MIT Press. 1965

[10] Chomsky N., *On Binding*, Linguistic Inquiry 11, 1980, pp.1-46

[11] Chomsky N., *Syntactic Structures*, Mouton, 1957

[12] Clocksin W. F., Mellish C. S., *Programming in Prolog*, Springer-Verlag, 1984

[13] Dale R., Mellish C., Zock M., *Current Research in Natural Language Generation*, Academic Press Limited, 1990

[14] Danlos L., *The Linguistic Basis of Text Generation*, Cambridge University Press, 1987

[15] Fillmore C., *The Case for Case*, in *Universals in Linguistic Theory*, Bach, Emmon, Harms, Roberts (eds.), New York: Holt, Rinehart, and Winston, 1968, pp. 1-88

[16] Fillmore C., *The Case for Case Reopened*, in *Syntax and Semantics - VIII: Grammatical Relations*, Cole and Sodock (eds.), New York: Academic, 1977

[17] Fries C. C., *The Structure of English*, Harcourt, Brace and World, 1952

[18] Hermanwekker , Haegeman L., *A Modern Course in English Syntax*, Routledge, London, 1985

[19] Hirst G., *Semantic Interpretation and the Resolution of Ambiguity*, Cambridge University Press, 1987

[20] Kaplan R. M., Bresnan J., *Lexical-Functional Grammar: A Formal System for Grammatical Representation*, in *The Mental Representation of Grammatical Relations*, ed. Joan Bresnan, The MIT Press, Cambridge, 1985, pp.173-281,

[21] Katz J. J., Fodor J. A., *The Structure of a Semantic Theory*, Language 39, pp.170-210.

[22] Kempson R. M., *Semantic Theory*, Cambridge University Press, 1979

[23] King M. ed., *Parsing Natural Language*, Academic Press Inc., 1983

[24] Langacker R.W., *Language and its Structure: Some Fundamental Linguistic Concepts*, Harcourt, Brace and World, Inc., 1968

[25] Langendoen D.T., Postal P.M., *The Vastness of Natural Languages*, Basil Blackwell Publisher Limited, 1984

[26] Maas H.D., *The MT System SUSY*, paper presented at The ISSCO Tutorial on Machine Translation, 1984

[27] Martin Atkinson, Kilyby D., Roca I., *Foundations of General Linguistics*, Unwin Hyman, London, 1988

[28] Mastermman M.,*Semantic Message Detection for MT, using an Interlingua*, Proc. 1961 Intl Conference on Machine Translation, 1961 pp. 438-475

[29] Melby A., *On Human-Machine Interaction in Translation*, in *Machine Translation: Theoretical and Methodological Issues*, ed. Sergei Nirenburg, The Cambridge University Press, Cambridge, 1987, pp.145-154

[30] Nagao M., *Role of Structural Transformation in a Machine Translation System*, in *Machine Translation, Theoretical and Methodological Issues*, ed. Sergei Nirenburg, Cambridge University Press, 1987, pp.262-277

[31] Nirenburg S.; *Knowledge and Choices in Machine Translation*, in *Machine Translation, Theoretical and Methodological Issues*, ed. Sergei Nirenburg, Cambridge University Press, 1987 , pp.1-21

[32] *Quintus Prolog User Reference Manual*

[33] Quirk R., Greenbaum S., Leech G., Svartvik J.,*A Comprehensive Grammar of the English Language*, Longman, London, 1985

[34] Roberts P., *Pattern of English*, Harcourt, Brace and World Inc., 1956

[35] Schank R., Bonnie L., Nash-Webber eds, *Theoretical Issues in Natural Language Processing*, Association for Computational Linguistics 1975

[36] Sowa J. F., *Conceptual Structures: Information Processing in Mind and Machine*, Addison-Wesley Publishing Company, Inc. 1984

[37] Tennant H., *Natural Language Processing - An Intorduction to an Emerging Technology*, Petrocelli Books, Inc. 1981

[38] Toma P., *SYSTRAN as a Multi-Lingual Machine Translation System* in *Commission of European Communities: Overcoming the Language Barrier*, Munich: Dokumentation Verlag, 1977, pp. 129-160

[39] Tsujii J., *Machine Translation: Future Aspects*, in *Language and Artificial Intelligence*, M. Nagao ed., North Holland, 1987, pp.265-282

[40] Tucker A. B.,*Current Strategies in Machine Translation Research and Development*, in *Machine Translation, Theoretical and Methodological Issues*, ed. Sergei Nirenburg, Cambridge University Press, 1987, pp.2-41

[41] Welch C., *The Sense of Language*, Martinus Nijhoff, Hague, 1973

[42] Wilks Y.A., *Grammar, Meaning and the Machine Analysis of Language*, Routledge and Kegan Paul Ltd., 1972

[43] Wilks Y., *Machine Translation and Artificial Intelligence*, in *Translating and the Computer*, Snell B.M. ed., North-Holland Publishing Company, 1979, pp.27-43

[44] Young D.J., *The Structure of English Clauses*, Hutchinson, London, 1980

Appendix F

Sample Interaction with ANGLABHARTI

----------SWITCHED OVER TO DEMO MODE ----------

```
your sentence
i am going to the market .
nAn kadai -ikku pOgirEn .

your sentence
i came by bus .
nAn Urthi -Al/-il vandEn .

your sentence
pigeons eat worms .
purAkkal puzhukkal -ai/-NULL/-irkku/enru sAppidugirana .

your sentence
they wandered-about in sheepskins and goatskins .
avargal sheepskins matrum Attu_thOlgal -il sutri_thirindanar .

your sentence
she is beautiful !
aval azhagAga irukkirAl !
```

your sentence

the lady , whose bag was stolen , was furious .

enda pen -udiaya pai thirudappattadu -0 anda pen kObamAga irundAl .

your sentence

the beautiful dog quickly bit the man .

azhagAna nAi manidan -ai/-NULL/-irkku/enru vEgamAga kadithadu .

your sentence

he sat at his desk .

avan avanadu mEjai -il utkArndAn .

your sentence

it is in the desk-drawer .

adu mEjaiyin_drawer -il irukkiradu .

your sentence

that desk is dusty .

anda mEjai azhukkAga irukkiradu .

your sentence

where is the desk ?

mEjai engE irukkiradu ?

your sentence

i saw the tall man .

nAn uyaramAna manidan -ai/-NULL/-irkku/enru pArthEn .

your sentence

the poor go away !

Ezhai_makkal dUramAga pOgirArgal !

your sentence

i can fish .

nAn mIn -ai/-NULL/-irkku/enru bOttilil_adaikkirEn .

nAn mIn_piddikka_vallavan .

your sentence

the rich went away !

panakkAra_makkal dUramAga pOnArgal !

your sentence

the rich man talks proudly !

panakkAra manidan garvathOdu pEsugirAn !

your sentence

he is in the house .

avan vIdu -il irukkirAn .

your sentence

he is in school .

avan pallikUdam -il irukkirAn .

your sentence

there is something pleasing about him !

avan -ai patri Edo onru sandOsham_alikkakUdiya_vagaiyil irukkiradu !

your sentence

the girl gave the boy a toy .

pen paiyan -ai/-ikku oru bommai -ai/-NULL/-irkku/enru kodUthAl .

your sentence

the girl asked the boy a toy .

pen paiyan -ai/-ikku oru bommai -ai/-NULL/-irkku/enru kEttAl .

your sentence
the girl gave a toy to the boy .
pen paiyan -ikku oru bommai -ai/-NULL/-irkku/enru koduthAl .

your sentence
the room has a large window , which faces south .
arai therku -ai/-NULL/-irkku/enru pArkkinra oru periya jannal
-ai/-NULL/-irkku/enru kondirukku .
arai pArkkinra oru periya jannal
-ai/-ikku therku -ai/-NULL/-irkku/enru kondirukku .
arai pArkkinra oru periya jannal
-ai/-ikku therku -ai/-NULL/-irkku/enru kondirukku .

your sentence
the bus , which runs south , is going to the market .
therku -ai/-NULL/-irkku/enru Oduginra Urthi kadai -ikku pOgiradu .

therku -ai/-NULL/-irkku/enru Oduginra Urthi kadai -ikku pOgiradu .

your sentence
the tall girl standing in the corner , who became angry
because you knocked-over her glass , when you entered ,
is my sister .
nI ullE_vandAi appOdu nI avaladu tumbler
-ai/-NULL/-irkku/enru thalli_vittAi Enra kAranathAl kObamAga Ana
Oram -il nirkinra uyaramAna pen enadu akkA .

your sentence
i told her to see him .
avan -ai/-NULL/-irkku/enru pArkka nAn aval -ai/-NULL/-irkku/enru kUrinen .

your sentence

he was believed to have been seen by her .

aval -Al/-il pArkkappattadAga avan nambappattAn .

your sentence

the man said she is beautiful .

aval azhagAga irukkirAl enru manidan kUrinAn .

your sentence

he , his two brothers , and sister were sent to live for three years
with their grandmother in the hut near the river .

nadi -arugE kudisai -il avargaladu pAtti -udan_kUdiya/-udan
mUnru varudangal -ikku vAzha avan , avanadu irandu annan_thambimArgal
matrum akkA anuppap_pattanar .

your sentence

they proved that he was wrong .

anda avan thavarAga irundAn enru avargal nirUbittArgal .

avan thavarAga irundAn enru avargal nirUbittArgal .

your sentence

away ran he !

dUramAga OdinAn avan !

your sentence

eating greedily is bad manners .

pErAsai_yOdu sAppiduvadu ketta pazhakkam .

your sentence

the sun keeps us very warm .

sUriyan nAm -ai/-NULL/-irkku/enru migavum idamAga vaithirukkiradu .

your sentence

never call a man a fool .

oru manidan -ai/-ikku oru muttAl -ai/-NULL/-irkku/enru eppOdumE kUppidAde .

your sentence

i came to the top floor of a house

in the corner of the old square

behind the church .

nAn mAdA_kOil -in_pinnE pazhaiya saduram/sandippu

-in/-ai_patriya Oram -il oru vIdu -in/-ai_patriya

mEl tharai/mAdi -ikku vandEn .

your sentence

----------SWITCHED OVER TO INTERACTIVE MODE----------

Please try again

yes